

WHAT IS CLAIMED IS:

1           1. A method of texturing a pixel, the method comprising:  
2           storing a texture argument in a general purpose register of a register file;  
3           issuing a texture command to a texture request buffer, wherein the texture  
4           command is associated with the texture argument;  
5           retrieving the texture command from the texture request buffer;  
6           retrieving the texture argument from the general purpose register;  
7           executing the texture command to produce a final texture value; and  
8           storing the final texture value in the register file.

1           2. The method of claim 1, wherein the final texture value is stored in the  
2           general purpose register of the register file, thereby overwriting the texture argument.

1           3. The method of claim 1, wherein the final texture value is stored in a  
2           second general purpose register of the register file.

1           4. The method of claim 1, wherein the texture command includes a  
2           texture parameter.

1           5. The method of claim 4, wherein the texture parameter specifies a  
2           texture mapping type.

1           6. The method of claim 4, wherein the texture parameter specifies a  
2           texture map.

1           7. The method of claim 4, wherein retrieving the texture argument  
2           comprises identifying the general purpose register from the texture parameter.

1           8. The method of claim 7, wherein the texture parameter specifies the  
2           general purpose register as a destination register to store the final texture value.

1           9. The method of claim 7, wherein the texture parameter specifies the  
2           general purpose register storing the texture argument and a second general purpose register to  
3           store the final texture value.

1               10.     The method of claim 1, further comprising locking the general purpose  
2 register to prevent changes to the texture argument in response to issuing a texture command;  
3 and

4               following storing the final texture value, unlocking the general purpose  
5 register.

1               11.     The method of claim 1, wherein the register file and the texture request  
2 buffer are included in a first execution unit, and wherein the method further comprises:  
3               repeating the steps of storing a texture argument, issuing a texture command,  
4 retrieving the texture command, retrieving the texture argument, executing the texture  
5 command, and storing the final texture value for a second execution unit including a second  
6 register file and a second texture request buffer.

1               12.     The method of claim 1, wherein the texture argument includes a set of  
2 texture coordinates.

1               13.     A graphics processing subsystem, comprising:  
2               an execution unit comprising a texture request buffer and a register file, the  
3 register file including a plurality of general purpose registers, wherein the execution unit is  
4 adapted to issue a texture command to the texture request buffer and to store a texture  
5 argument in the register file;  
6               a texture unit adapted to read the texture command from the texture request  
7 buffer and to retrieve the texture argument from the register file; and  
8               wherein the texture unit is further adapted to retrieve a portion of a texture  
9 map in response to the texture command, to compute a final texture value from the portion of  
10 the texture map, and to store the final texture value in a destination register of the plurality of  
11 registers of the register file.

1               14.     The graphics processing subsystem of claim 13, wherein the execution  
2 unit is adapted to store the texture argument in the destination register of the plurality of  
3 registers, such that the texture unit is adapted to overwrite the texture argument in storing the  
4 final texture value.

1               15.     The graphics processing subsystem of claim 13, wherein the execution  
2 unit is adapted to store the texture argument in a source register of the plurality of registers,  
3 wherein the source register is separate from the destination register.

1               16.     The graphics processing subsystem of claim 13, wherein the texture  
2 command includes a texture parameter.

1               17.     The graphics processing subsystem of claim 16, wherein the texture  
2 parameter specifies a texture mapping type.

1               18.     The graphics processing subsystem of claim 16, wherein the texture  
2 parameter specifies a texture map.

1               19.     The graphics processing subsystem of claim 16, wherein the texture  
2 unit is further adapted to determine from the texture parameter a source register of the  
3 plurality of registers storing the texture argument.

1               20.     The graphics processing subsystem of claim 19, wherein the texture  
2 parameter specifies that the destination register of the plurality of registers is also the source  
3 register of the plurality of registers.

1               21.     The graphics processing subsystem of claim 19, wherein the texture  
2 parameter specifies the source register and the destination register of the plurality of registers,  
3 wherein the source register is separate from the destination register.

1               22.     The graphics processing subsystem of claim 13, wherein the execution  
2 unit is further adapted to lock the destination register to prevent changes to the texture  
3 argument in response to issuing a texture command and to unlock the destination register in  
4 response to the texture unit storing the final texture value in the destination register.

1               23.     The graphics processing subsystem of claim 15, wherein the execution  
2 unit is adapted to lock the source register to prevent changes to the texture argument in  
3 response to issuing a texture command and to unlock the source register in response to the  
4 texture unit retrieving the texture argument from the register file.

1               24.     The graphics processing subsystem of claim 13, further comprising:

2                   a second execution unit comprising a second texture request buffer and a  
3   second register file, the second register file including a plurality of general purpose registers,  
4   wherein the second execution unit is adapted to issue a second texture command to the  
5   second texture request buffer and to store a second texture argument in the second register  
6   file; and

7                   wherein the texture unit is further adapted to read the second texture command  
8   from the second texture request buffer, to retrieve the second texture argument from the  
9   second register file, to retrieve a portion of a second texture map from the texture memory in  
10   response to the second texture command, to compute a second final texture value from the  
11   portion of the texture map, and to store the second final texture value in a destination register  
12   of the plurality of registers of the second register file.

1                 25.   The graphics processing subsystem of claim 24, further comprising:  
2                   a multiplexer switch adapted to alternately connect the first execution unit and  
3   the second execution unit with the texture unit.

1                 26.   The graphics processing subsystem of claim 25, wherein the texture  
2   unit is connected with the first execution unit and the second execution unit according to a  
3   round robin schedule.

1                 27.   The graphics processing subsystem of claim 25, wherein the texture  
2   unit is connected with the first execution unit and the second execution unit according to a  
3   priority function adapted to maximize the performance of the graphics processing subsystem.

1                 28.   The graphics processing subsystem of claim 13, further comprising a  
2   texture memory connected with the texture unit and adapted to store a texture map.